A New Predictive Analyzer of English

Hiroyuki Musha

Department of Information Science
Tokyo Institute of Technology
Ohokayama, Meguro-ku, Tokyo 152, JAPAN

ABSTRACT

Aspects of syntactic predictions made during the recognition of English sentences are investigated. We reinforce Kuno's original predictive analyzer[1] by introducing five types of predictions. For each type of prediction, we discuss and present its necessity, its description method, and recognition mechanism. We make use of three kinds of stacks whose behavior is specified by grammar rules in an extended version of Greibach normal form. We also investigate other factors that affect the predictive recognition process, i.e., preferences among syntactic ambiguities and necessary amount of lookahead. These factors as well as the proposed handling mechanisms of predictions are tested by analyzing two kinds of articles. In our experiment, more than seventy percent of sentences are recognized and looking two words ahead seems to be the critical length for the predictive recognition.

1. Introduction

When human reads normal sentences, we rarely feel something is wrong with the structure we are constructing and are seldom compelled to backtrack for reconstructing an alternative. If we could simulate the internal mechanism that makes it possible to select deterministically the unique syntactic structure in a simple way, we may be able to construct more natural and efficient language processing systems. In this paper, we focus our attention on syntax of natural languages, particularly English, and predictions or expectations that can be made solely with syntactic information during the sentence recognition process are analyzed in detail. It includes machine executable mechanisms that enable proper handling of analyzed aspects and a description method of the mechanisms as grammar rules. The recognition method can be seen as a deterministic one [2] if we permit looking some words ahead. Also included in this paper are results of an experimental analysis in which more than seventy percent of sentences are recognized.

An analyzer which gives special attention to predictions was once developed by Kuno [1]. The analyzer makes use of the simple stack mechanism whose behavior is specified by rules described in Greibach normal form. In the method, however, we can find several kinds of rules that do not correspond to human predictive recognition process, which will be pointed out in this paper.

The following discussion is based mainly on the author's (subjective) retrospect of the recognition process of English sentences. The author's mother tongue is Japanese and he has been learning English as a second language. It seems to the author that he can understand better how he recognizes English than how he recognizes Japanese since he has been learning English consciously and can observe rather objectively the process of recognition.

The rest of this paper is organized as follows. In the next section, we discuss aspects of predictions, laying stress upon their proper handling by computers. The following section presents the results of an experiment. The conclusions are presented in the last section.

2. Aspects of Predictions

While reading or hearing English, we constantly predict or expect what may follow next. Such predictions can be classified into six types which we will describe below.

2.1. Essential Predictions

The simplest type of prediction, which forms the basis of the following discussions, is presented in this subsection. The characteristic of this type of prediction is that it is essential in forming an acceptable sentence structure.

Phrase structure grammar rules, especially those in Greibach normal form, can naturally describe this kind of prediction: we can consider the terminal symbol (or the lexical category) on the right-hand side of a rule as the current word and the nonterminal symbols that follow the terminal symbol as new predictions [3]. For example, the following rule describes what we predict when we encounter a transitive verb at the beginning of a verb phrase.

VP → vt NP

Note that the new prediction, NP, is essential to form a verb phrase. By adopting this kind of rules as a means of structural description of sentences, we can easily capture the structures by using the stack mechanism [1].

In the following subsections, except for the last subsection, these rules and the mechanism are gradually reinforced in order to handle a newly introduced prediction type. The extended mechanism provides us with a simpler (yet still powerful) means for recognition of sentence structures than, for example, ATN framework [4]. Other factors that affect the predictive recognition process are discussed in the last subsection.

2.2. Optional Predictions

We now extend our recognition mechanism by introducing optional predictions. This type of prediction is needed to handle postpositional modifiers that are not essential to form a sentence.

In the previous subsection, we saw that rules in Greibach normal form are suitable for expressing our predictive recognition process, but any rule should not predict too much. Consider the following rule that explains a possible structure of noun phrases.

NP → article NP-ART ADJ_CLAUSE

Concerning the correspondence with human language understanding process, however, the rule cannot be considered a good simulation of our understanding process: we predict a postpositional modifier, like an adjectival clause, not at the beginning of a noun phrase but at the beginning of the modifier. For our purpose, therefore, we must exclude this kind of rule that do not express our predictions properly.

Optional predictions are used to capture these structures. Here, we also extend the rule description to keep the correspondence between the grammar rules and the recognizing mechanism: we introduce the *shifting flag*. The following rules are used to capture postpositional modifiers.

	CW	**CP**		**SF**	**NPr**
(1)	art	NP	→	t	NP-ART
(2)	noun	NP-ART	→	t	*NP-N
(3)	rel_pro	NP-N	→	nil	ADJ_CLAUSE

The first rule, for example, can be interpreted as follows: **IF** the current word (**CW**) is an article and the current prediction (**CP**: the top element of the stack) is NP, **THEN** shift the current word pointer (since the shifting flag (**SF**) is *t*) and replace the current prediction by the new prediction (**NPr**).

The shifting flag enables us to proceed two or more state changes while looking at a single word. By using these notations

and the rules we can specify the state changes of the stack as shown in Figure 2-1. The prediction NP-N, with a prefix '*' which shows it is optional, is interpreted as the state in which a noun essential to form a noun phrase has already appeared and it may end there. It will be popped out from the stack or will be replaced by a new prediction according to the word that follows.

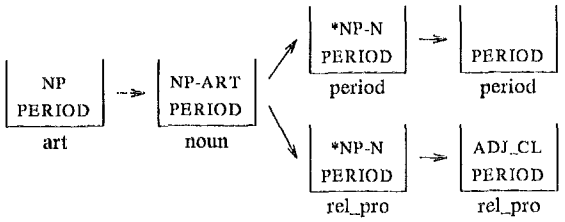

Figure 2-1. Handling of the optional prediction.

2.3. Bunch Predictions

We extend our model by introducing bunch predictions which enable us to predict a set of syntactic categories simultaneously. In the following subsection, we see that this kind of prediction is useful for handling coordinate conjunctions, too.

Various kinds of syntactic units can follow the verb *be* in a verb phrase and we cannot selectively predict one of these possibilities when we are reading the words, such as *am* or *were*, etc. The bunch predictions we introduce enable us to cope with this kind of predictions.

The following rule shows how to write a bunch prediction in a rule.

(be fnt) (VP fnt) → t (bunch (NP) (ADJ) ((VP ing))] *VP_MOD

When a bunch prediction is pushed onto the stack, it works as if it were a single prediction until it becomes the top of the stack, and one of the constituent of the bunch prediction is, then, chosen to be appropriate according to the word encountered.

2.4. And Stack

In this subsection, we introduce another stack called the *and stack* to handle coordinate conjunctions. The method described here resembles that in [5] or [6], but with the *and stack* we can handle them quite simply.

The appearance of coordinate conjunctions are usually not predictable and it triggers a new kind of operation. Let us consider the following sentences.

(1) *Mary had a little lamb and a kitten.*
(2) *Mary had a little lamb and washed him every day.*
(3) *Mary had a little lamb and she was always with him.*

Conventional phrase structure grammar rules like:

S → S and S

are not directly useful for predictive recognition of the sentences. The structure that follows *and* depends not on the word itself but on the proceeding syntactic units being constructed. In the above sentences, a noun phrase, a verb phrase, and a clause are being constructed before the word, and each of these categories reappears in each of the three sentences, respectively.

By using the *and stack*, we can easily recognize these structures. Figure 2-2 shows the relationship between the *prediction stack* (the stack that holds predictions) and the *and stack* where unnecessary details are omitted. At stage (ii), the first prediction is replaced by two predictions NP and VP only by looking the first word *Mary*. The lower element of the *and stack* is changed to (VP S), which shows that while the VP of the *prediction stack* is being processed, we are constructing both VP and S. In the same way, the stacks change their states as shown in the figure and a list (NP VP S) is made and pushed on the *and stack* when we reach the word *and*. The only thing we have to do is that we make a bunch prediction [bunch (NP) (VP) (S)] and replace *NP-N by the bunch prediction. By looking at the words that follow we can choose one of the constituent predictions of the bunch prediction and process the rest of the sentence.

Note that the following sentence can also be recognized by this strategy:

Mary looked for and found the unicorn.

A list (NP VP S) has been built when we encounter the conjunction, and VP is used to capture the structure of the rest of the sentence.

The following rule description is used to trigger the above explained operation:

and ?P → t (special and_stack)

where ?P indicates that applicability of the rule does not depend on the current prediction.

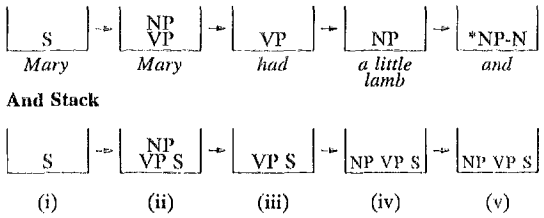

Figure 2-2. Relation between *and stack* and *prediction stack*.

2.5. Insertive Structures

Some kinds of words trigger insertive structures which are usually not predicted, and cause a kind of suspension of construction of structures being built. Some adverbs, prepositional phrases and adverbial phrases and clauses are such structures. Here are three examples, where we use a pair of quotes to distinguish insertive structures.

(1) *There are economic risks and "generally" a lack of available data.*
(2) *He adapted "for linguists" an existing system of formalization.*

In order to express insertive structures, we use the following notation.

(A-1) adverb ?P → t
(A-2) preposition ?P → nil PP

These rules are applicable for almost all old predictions provided that the current word belongs to the **CW** part of the rules. In this case, however, the top element of the *prediction stack* will not be popped. The new prediction(s), if they exist, will be pushed onto the current prediction.

For example, (2) will be processed as shown in Figure 2-3. At first, the object noun phrase, NP, of the verb "adapted" is predicted. The rule (A-2) is then applied and the recognition of NP is suspended until the prepositional phrase is recognized by the prediction PP.

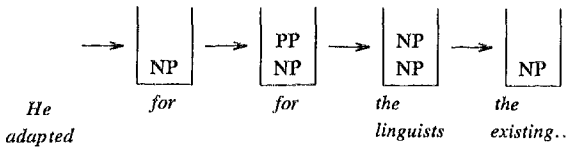

Figure 2-3. Handling of insertive structures.

2.6. NP stack

We introduce yet another stack, the *NP stack*, to handle structures where a noun phrase is missing, e.g. relative clauses. This approach is widely used, e.g. in [7]. The fact is that people do not handle these structures in a totally different way comparing with normal clauses. It seems that when we encounter a relative pronoun, we push a noun phrase onto a kind of stack, which we call the *NP stack*, and pop it when it is needed to fill out the gap afterwards.

The following rule is used to simulate the above operations.

rel_pro ADJ_CLAUSE → nil +S

The prefix '+' of the new prediction indicates that we push a noun phrase onto the *NP stack*.

2.7. Looking ahead and Preference

In this subsection we discuss necessity of looking ahead and preference among syntactic ambiguities that affect the predictive recognition process.

Some sort of lookahead facility is necessary to reflect the delay in making syntactic structure of sentences. In sharp contrast with Marcus's deterministic parser [2], we only make use of a word as the unit of lookahead.

In the middle of a sentence we usually do not look back to see what the preceding structure was in order to build up a dominating structure. In Marcus's parser, however, we can make a rule like: "**IF** the first element is NP and the second element is VP, **THEN** let NP and VP be sons of S," where NP which was recognized some time before is referenced again. This framework seems to be too strong as a simulation of our internal process. The approach taken in this research is to permit more appropriate and generalized predictions as described in the previous subsections.

In our experiment, we make use of lookahead by permitting backtracking within a limited range: once the analyzer reached the n-th word, it would not cancel the previous decision made when it was processing (n-k)-th word, where k is the length of lookahead. The necessary length of the lookahead is investigated in the experiment.

Currently, preference factors are treated in the following manner. The syntactic categories a word belongs to are linearly ordered. Grammer rules are divided into two groups, usual and unusual: the rules that trigger insertive structures with some other uncommon rules are included in the latter group. Although the strategies are not fixed, generally we try each syntactic category one by one according to the order induced, and the usual rules are tried before unusual ones.

3. Experiment

The mechanisms described in the previous section were tested by analyzing two kinds of articles. The articles used in the experiment were a manual of a computer software and an abstract article on world economics. At first, basic grammar rules were written and they were revised and reinforced by looking at the result of the previous analysis.

The output of the analyzer is a kind of tree structure as shown in Figure 3-1.

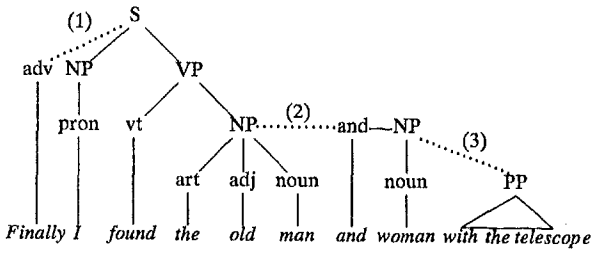

Figure 3-1. Tree structure constructed by the analyzer.

Structures that are captured by optional predictions and predictions made through the *and stack* or insertive rules are called pending structures. In the example, (1), (2) and (3) are pending structures: (1) is recognized as an insertive structure; (2) is captured through the *and stack*; and (3) is captured by an optional prediction. As shown in the figure, we temporarily attach them to the preceding predictions. In this representation, the word *woman* is modified by the prepositional phrase *with the telescope*. We, however, can easily obtain other plausible sentence structures. For example, if we attach (3) to VP, we get a tree structure where the prepositional phrase modifies the verb phrase. In our experiment, a sentence is said to be successfully recognized if we can get an appropriate tree structure by moving pending structures (if necessary).

The success rate and its relation with the length of lookahead was as follows. Of the 85 sentences from each article, 65 (manual) and 70 (abstract) of them were analyzed as desired by making use of looking two words ahead, the current and the next word, while only one additional success was reported on each article by looking one more word ahead.

4. Conclusion

Based on the observation of human recognition process of English sentences of a non-native speaker, predictions we make during the process are analyzed. We have also presented a description method of such predictions as grammar rules which is based on Greibach normal form, and recognition mechanisms that are specified by these rules, realized by using three stacks: the *prediction stack*, the *and stack*, and the *NP stack*. The extension of the rule description and introduction of these stacks provide us with a simple yet powerful means for recognition of syntactic structures.

An experimental analysis of more than 150 sentences is carried out, and necessary length of lookahead and preference factors as well as the plausibility of the above mechanisms are tested. Over 70 percent of the sentences are recognized as desired and looking two words ahead seems to be the critical length for the predictive recognition.

Acknowledgements

I would especially like to thank my adviser, Prof. A. Yonezawa of Tokyo Institute of Technology, for his valuable comments on this research and encouragement. I also thank the members of Yonezawa Lab. for their comments on my research. I also give my special thanks to the managers of Resource Sharing Company who allowed me to use their valuable dictionary for my research.

References

[1] S. Kuno, "The Predictive Analyzer and a Path Elimination Technique," *Comm. ACM, Vol. 8*, pp. 453-462, 1965.

[2] M. P. Marcus, *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge, 1980.

[3] M. Nagao, *Language Engineering*, Shoko-do, Tokyo, 1983, (in Japanese).

[4] W. A. Woods, "Transition Network Grammars for Natural Language Analysis," *Comm. ACM, Vol. 13*, pp. 591-606, 1970.

[5] T. Winograd, *Understanding Natural Language*, Academic Press, New York, 1972.

[6] B. K. Boguraev, "Recognising Conjunctions within the ATN Framework," pp. 39-45, in *Automatic Natural Language Parsing* (K.S. Jones and Y. Wilks, eds.), Ellis Horwood limited, 1983.

[7] T. Winograd, *Language as a Cognitive Process, Vol. 1: Syntax*, Addison-Wesley, 1983.